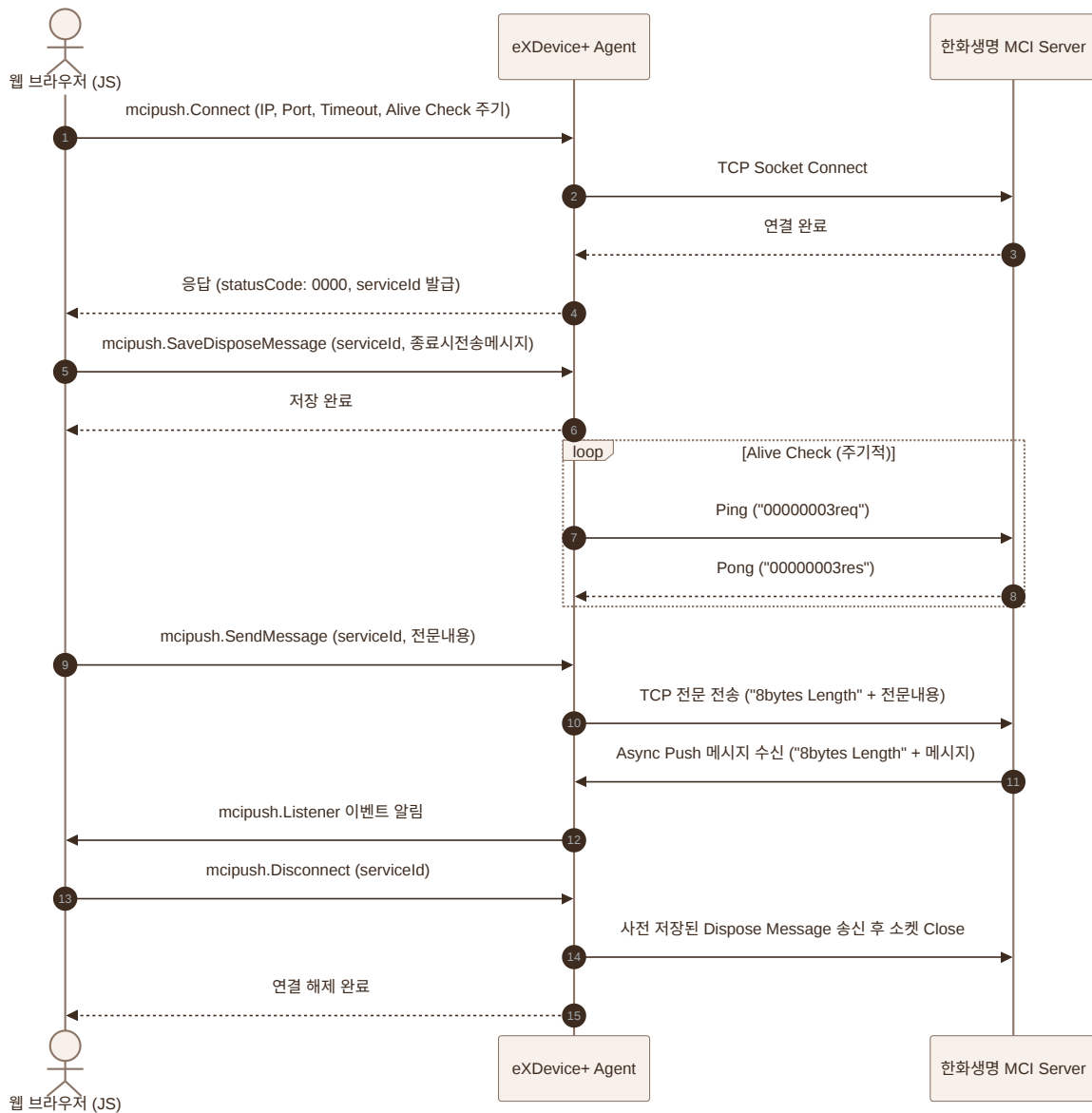


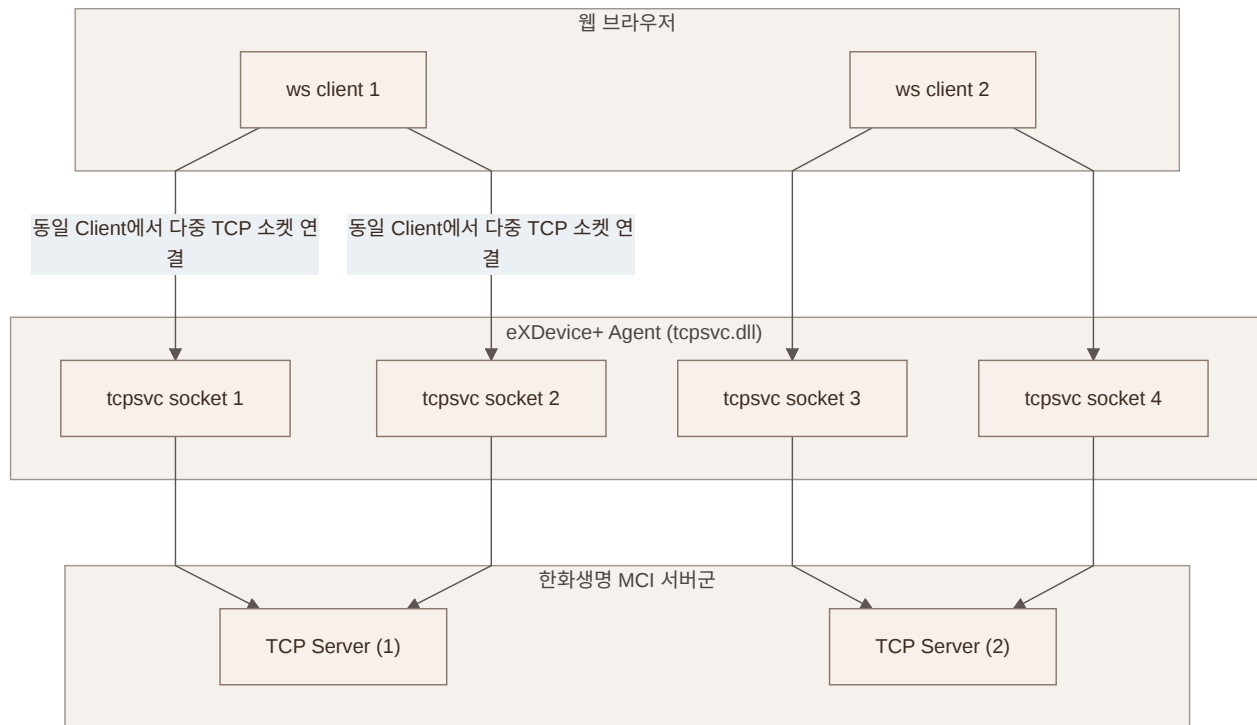
MCI Server (TCP) 통신 서비스

eXDevice+ Agent를 중계 서버로 활용하여 웹 브라우저가 외부 한화생명 MCI(Message Communication Interface) 서버와 직접 TCP 소켓 통신을 수행할 수 있도록 지원하는 전용 프로토콜 서비스입니다.



1. 통신 아키텍처 및 다중 소켓 접속 구조

tcpsvc.dll 은 하나의 웹소켓 클라이언트(브라우저)에서 다중 TCP 서버와의 소켓 연결을 지원하며, 단일 웹소켓 연결 상에서 동일한 TCP 서버에 대해 2개 이상의 다중 소켓 세션을 동시에 수립하고 독립적으로 제어할 수 있습니다.



2. TCP 패킷 통신 프로토콜 (Packet Frame Format)

eXDevice+와 한화생명 MCI 서버 간의 TCP 데이터 송수신은 "고정 8바이트 메시지 길이 헤더(Length)" + "가변 전송 메시지 본문(Body)" 규격을 준수합니다.

Length (고정 8 Bytes)	Body (가변 길이 전송 메시지 본문)
0 0 0 0 0 0 1 5	T o S e n d M e s s a g e

- **헤더 규격:** 전송할 메시지 본문의 바이트 길이를 8자리 10진수 문자열로 포맷팅합니다. (예: 메시지 길이가 15바이트인 경우 00000015를 헤더로 결합하여 총 23바이트를 전송)
- **수신 검증:** TCP 서버로부터 수신한 초기 8바이트 데이터가 정수형(Integer) 값이 아닌 경우, eXDevice+는 즉시 TCP 소켓 연결을 해제하고 웹소켓 클라이언트에게 상태코드 3(전문 길이 포맷 오류)을 통보합니다.
- **라이프사이클 연동:** 서비스를 요청한 웹소켓 클라이언트(브라우저)와 eXDevice+ 간의 연결이 해제 되면, 해당 클라이언트가 생성했던 모든 TCP 서버와의 소켓 연결도 즉시 일괄 해제됩니다.

3. Keep-Alive (Ping-Pong) 동작 메커니즘

- **Ping 전송:** eXDevice+는 `pingInterval` 주기에 맞춰 MCI 서버로 `"req"` 문자열(8바이트 헤더 포함: `"00000003req"`)을 전송합니다.
- **Pong 응답:** MCI 서버는 Ping 수신 시 `"res"` 문자열(8바이트 헤더 포함: `"00000003res"`)을 회신해야 합니다.
- **비정상 단절 처리:** `pingTimeout` 시간 내에 `"res"` 응답을 수신하지 못한 경우, eXDevice+는 소켓 연결을 즉시 강제 해제합니다. (이 경우 `DisposeMessage` 는 서버로 전송하지 않습니다.)

4. 브라우저 TCP 세션 식별 키 관리

하나의 웹소켓 연결에서 여러 TCP 소켓을 식별하고 제어하기 위해 브라우저는 다음 두 가지 키값을 관리해야 합니다.

No.	Key 명칭	확인 가능 시점	주요 용도 및 역할
1	requestKey	브라우저에서 <code>mcipush.Connect</code> 호출 시 생성한 키	서버로부터 비동기 <code>mcipush.Listener</code> 푸시 메시지 수신 시 어떤 TCP 서버/소켓에서 온 메시지인지 식별하기 위한 식별자
2	servicId	<code>mcipush.Connect</code> 호출 성공 시 응답받은 세션 ID	해당 TCP 소켓을 대상으로 추가 서비스 (<code>mcipush.SendMessage</code> , <code>mcipush.SaveDisposeMessage</code> , <code>mcipush.Disconnect</code>)를 요청할 때 식별자로 사용

서비스 리스트

No.	서비스명	API명	비고
1	MCI 서버 TCP 연결	<code>mcipush.Connect</code>	TCP 소켓 연결 및 <code>serviceId</code> 발급
2	Disconnect 메시지 등록	<code>mcipush.SaveDisposeMessage</code>	소켓 해제 시 자동 송신 메시지 등록
3	메시지 전송	<code>mcipush.SendMessage</code>	MCI 서버로 전문 송신
4	MCI Push 메시지 수신	<code>mcipush.Listener</code>	비동기 Push 수신 이벤트 (직접 호출 불가)
5	MCI 서버 연결 해제	<code>mcipush.Disconnect</code>	TCP 소켓 연결 종료

서비스 상태 코드 리스트

1. 서비스 응답 상태 코드 (`tcpsvcCode`)

`mcipush` 서비스 API 호출 결과 객체의 `return` 또는 `return.returnValue` 값 매핑표입니다.

리턴코드	내용	조치 가이드
0	성공	정상 처리되었습니다.
-1	정의되지 않은 예외	eXDevice+ 로그 파일을 확인하시길 바랍니다.
1	이미 연결됨	MCI Server와 이미 연결되어 있습니다. 기존 연결 해제 후 재연결을 수행하시길 바랍니다.
2	잘못된 IP 주소	유효하지 않은 IP 주소가 입력되었습니다. 입력한 IP 주소를 확인하시길 바랍니다.
3	잘못된 Port 번호	유효하지 않은 Port 넘버가 입력되었습니다. 입력한 Port 넘버를 확인하시길 바랍니다.
4	MCI 서버 연결 불가	MCI Server와 연결할 수 없습니다. MCI Server의 주소 또는 네트워크 연결을 확인하시길 바랍니다.
5	연결 종료됨	사용자/eXDevice+에 의해 MCI Server와의 연결이 종료되었습니다.
6	연결 타임아웃	MCI Server와 연결 중 타임아웃이 발생했습니다. MCI Server의 주소 또는 네트워크 연결을 확인하시길 바랍니다.
7	미연결 상태	MCI Server에 연결되어 있지 않습니다. <code>serviceId</code> 유효성 및 사전 연결을 확인하시길 바랍니다.

리턴코드	내용	조치 가이드
8	Disconnect 실패	MCI Server로부터 Disconnect에 실패했습니다.
9	메시지 전송 실패	MCI Server에 메시지 전송에 실패했습니다.
10	LogFormat 설정 오류	유효하지 않은 LogFormat이 입력되었습니다. <code>modules.json</code> 을 확인하시길 바랍니다.

2. 푸시 리스너 상태 코드 (`listenerCode`)

리턴코드	내용	상세 설명
0	성공	정상적으로 푸시 메시지를 수신했습니다.
-1	정의되지 않은 예외	로그 파일을 확인하시길 바랍니다.
1	연결 해제됨	MCI 서버와 연결이 해제되었습니다.
2	Pong 미수신	MCI 서버로부터 pong을 받지 못했습니다.
3	전문 길이 포맷 오류	MCI 서버로부터 수신한 Message length 데이터가 Integer 값이 아닙니다.

1. MCI 서버 TCP 연결

API명

- `mcipush.Connect`

정의

- 대상 MCI 서버의 IP와 Port로 TCP 소켓을 생성하고, Keep-Alive 스레드를 가동하며 리스너를 바인딩합니다.

호출 예시

```
{
  "service": "mcipush.Connect",
  "requestKey": "MCI SOCK_01",
  "param": {
    "serverIp": "127.0.0.1",
    "serverPort": "9480",
    "socketTimeout": "3000",
    "retry": "3",
    "pingTimeout": "3000",
    "pingInterval": "600000"
  }
}
```

- serverIp
 - 대상 한화생명 MCI 서버 IP 주소입니다.
- serverPort
 - 대상 한화생명 MCI 서버 TCP Port 번호입니다.
- socketTimeout
 - TCP 소켓 연결 수립 타임아웃 시간입니다. (밀리초 단위, ms)
- retry
 - 서버 연결 시도 시 또는 Ping 전송 후 Pong 수신 대기 시의 최대 재시도 횟수입니다.

- pingTimeout
 - Alive check(Ping) 전송 후 Pong 응답 수신 대기 타임아웃 시간입니다. (밀리초 단위, ms)
- pingInterval
 - Alive check(Ping) 주기입니다. (밀리초 단위, ms, 예: 600000 = 10분)

응답 예시

```
{
  "service": "mcipush.Connect",
  "requestKey": "MCI_SOCKET_01",
  "statusCode": "0000",
  "return": {
    "returnValue": 0,
    "serviceId": "0c089b51-083b-4654-8261-aa2220ebf809"
  }
}
```

- returnValue
 - 서비스 상태 코드입니다. (0: 성공, 그 외: 에러 코드)
- serviceId
 - 생성된 TCP 소켓 세션의 고유 식별자입니다. (이후 `SendMessage`, `SaveDisposeMessage`, `Disconnect` 호출 시 파라미터로 사용됩니다.)

2. Disconnect 메시지 등록

API명

- `mcipush.SaveDisposeMessage`

정의

- 웹브라우저와 eXDevice+ 간 웹소켓 연결이 끊어지거나, 웹브라우저에서 TCP 소켓 연결 해제 (`Disconnect`)를 요청할 때 소켓 종료 직전 MCI 서버로 자동 전송할 고별(Dispose) 전문을 사전에 eXDevice+ 메모리에 등록합니다.

호출 예시

```
{
  "service": "mcipush.SaveDisposeMessage",
  "requestKey": "REQ_SAVE_DISP_01",
  "param": {
    "serviceId": "0c089b51-083b-4654-8261-aa2220ebf809",
    "message": "DISCONNECT|CLIENT_SESSION_CLOSE"
  }
}
```

- `serviceId`
 - 대상 TCP 소켓 식별자입니다. (`mcipush.Connect` 응답으로 발급받은 값)
- `message`
 - 세션 종료 또는 연결 단절 시 MCI 서버로 자동 전송할 종료 전문 내용입니다.

응답 예시

```
{
  "service": "mcipush.SaveDisposeMessage",
  "requestKey": "REQ_SAVE_DISP_01",
  "statusCode": "0000",
  "return": 0
}
```

- return
 - 서비스 상태 코드입니다. (0: 성공, 그 외: 에러 코드)

3. 메시지 전송

API명

- `mcipush.SendMessage`

정의

- 수립된 TCP 소켓을 통해 한화생명 MCI 서버로 데이터를 송신합니다. (전송 시 8bytes 메시지 길이 헤더가 자동으로 결합되어 송신됩니다.)

호출 예시

```
{
  "service": "mcipush.SendMessage",
  "requestKey": "REQ_MCI_SEND_01",
  "param": {
    "serviceId": "0c089b51-083b-4654-8261-aa2220ebf809",
    "message": "TR001|20260901|REQ_LOAN_STATUS|CUSTOMER_001"
  }
}
```

- `serviceId`
 - 전송 대상 TCP 소켓 식별자입니다.
- `message`
 - MCI 서버로 전송할 전문 데이터 문자열입니다.

응답 예시

```
{
  "service": "mcipush.SendMessage",
  "requestKey": "REQ_MCI_SEND_01",
  "statusCode": "0000",
  "return": 0
}
```

- return
 - 서비스 상태 코드입니다. (0: 성공, 그 외: 에러 코드)

4. MCI Push 메시지 수신

API명

- `mcipush.Listener`

정의

- MCI 서버로부터 TCP 스트림을 통해 비동기 푸시 전문이 수신되었을 때, eXDevice+가 브라우저의 WebSocket 콜백 핸들러로 전달하는 이벤트 메시지입니다. (직접 호출 불가)

수신 메시지 예시 (브라우저 수신 형태)

```
{
  "service": "mcipush.Listener",
  "requestKey": "MCI SOCK_01",
  "statusCode": "0000",
  "return": {
    "returnValue": 0,
    "message": "TR001|RES_LOAN_STATUS|SUCCESS|BALANCE:5000000"
  }
}
```

- requestKey
 - `mcipush.Connect` 호출 시 전달했던 `requestKey`가 포함되어 전달되므로, 브라우저는 이 값을 통해 어떤 TCP 소켓 연결로부터 수신된 푸시 메시지인지 식별할 수 있습니다.
- returnValue
 - 리스너 상태 코드입니다. (0: 성공, 그 외: 에러 코드)
- message
 - MCI 서버로부터 수신된 비동기 푸시 전문 데이터 본문입니다.

5. MCI 서버 연결 해제

API명

- `mcipush.Disconnect`

정의

- 지정한 `serviceId`에 매핑된 TCP 소켓 연결을 종료합니다. 등록된 `DisposeMessage`가 존재하는 경우 TCP 서버에 Dispose 메시지를 전송하고 확인 메시지를 수신한 뒤 안전하게 TCP 소켓을 Close합니다.

호출 예시

```
{
  "service": "mcipush.Disconnect",
  "requestKey": "REQ_MCI_DISC_01",
  "param": {
    "serviceId": "0c089b51-083b-4654-8261-aa2220ebf809"
  }
}
```

- `serviceId`
 - 연결을 해제할 대상 TCP 소켓 식별자입니다.

응답 예시

```
{
  "service": "mcipush.Disconnect",
  "requestKey": "REQ_MCI_DISC_01",
  "statusCode": "0000",
  "return": 0
}
```

- `return`
 - 서비스 상태 코드입니다. (0: 성공, 그 외: 에러 코드)

